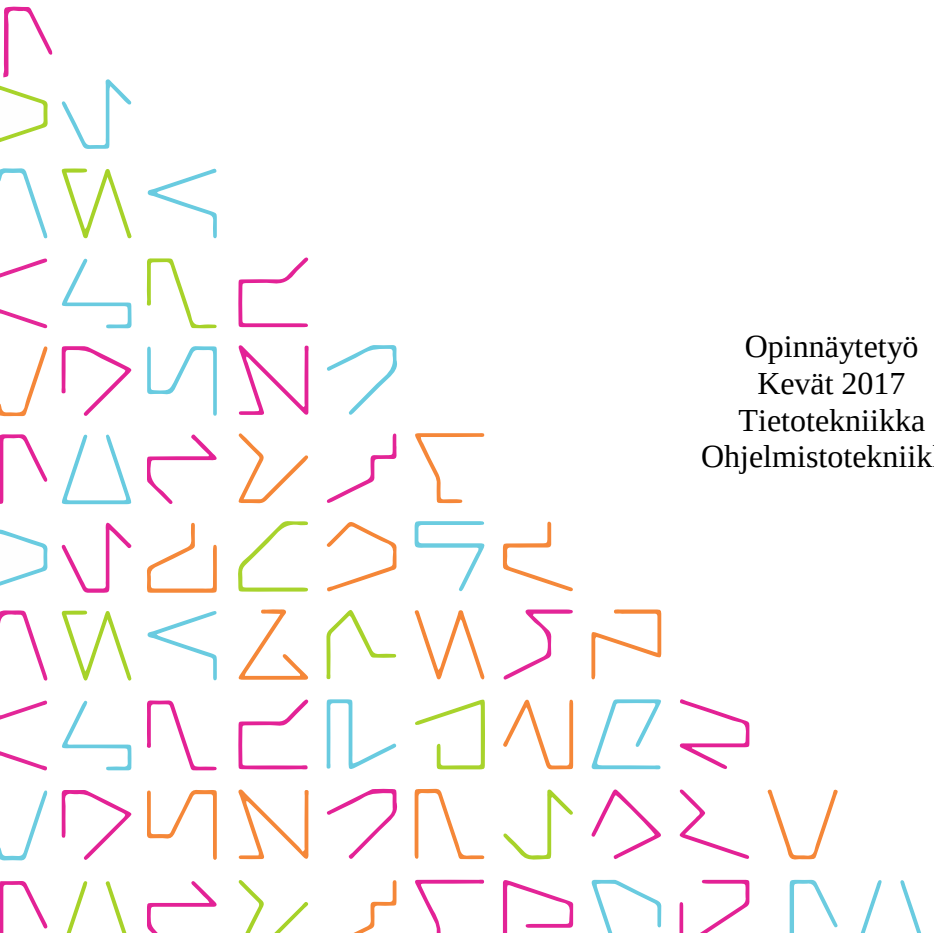
TAMPEREEN
AMMATTIKORKEAKOULU

MS Office -sisällönjulkaisulaajennus

Anssi Rantanen



Opinnäytetyö
Kevät 2017
Tietotekniikka
Ohjelmistotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka

RANTANEN ANSSI
MS Office -sisällönjulkaisulaajennus
Opinnäytetyö 27 sivua
Kevät 2017

Opinnäytetyön tarkoituksena oli tuottaa sisällönjulkaisulisäosat PowerPoint- ja Excel-sovelluksiin. Lisäosat tulivat osaksi tilaajan työnohjausjärjestelmää. Lisäosien tehtävänä on siirtää dataa Office-ohjelmista visualisointipalvelimelle, jossa se jatkokäsitellään.

Työssä määritellään lisäosien yhteiset- ja sovelluskohtaiset ominaisuudet. Näiden ominaisuuksien perusteella evaluoidaan teknisiä ratkaisutapoja ja-tekniikoita, joista valitaan parhaiten määritellyt ominaisuudet toteuttavat ratkaisutavat. Valittujen ratkaisujen avulla luodaan valmiit sovellusratkaisut.

Työssä kuvataan syntynyt ohjelmistoratkaisu ja sen rakenne, kehityksen aikana tapahtuneita muutoksia ja muutoksiin johtaneet syyt.

Avainsanat: MS Office sisällönjulkaisulaajennus

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
ICT Engineering
Software Engineering

RANTANEN ANSSI
MS Office Content Publishing Tools

Bachelor's thesis 27 pages
Spring 2017

Purpose of this thesis was to design and produce content publishing add-ins for PowerPoint and Excel programs. Add-ins were created to be used in a visualization and guidance system. Purpose of add-ins is to publish data from Office programs into visualization server, where data will be further refined.

Thesis defines general and add-in specific features. By using chosen features possible software solutions were selected, which were used to produce final software solution. Thesis describes final software solution by illustrating parts of architecture, changes made during production process.

Key words: MS Office content publishing tools

SISÄLLYS

1 JOHDANTO 7
2 KOKONAISUUDEN KUVAUS 8
2.1 Ongelma 8
2.2 Työn tehtävä 9
2.2.1 PowerPoint-lisäosa 9
2.2.2 Excel-lisäosa 9
3 TYÖKALUT JA KIELET 10
3.1 Visual Basic 10
3.2 Visual Basicin erityispiirteitä 10
3.3 Visual Studio Tools for Office -kehitystyökalut 11
4 VAATIMUKSET 12
4.1 Yleiset vaatimukset 12
4.1.1 Versiotuki 12
4.1.2 Autentikointi ja autorisointi 12
4.1.3 HTTP-protokolla 12
4.1.4 Tilan säilyttäminen 13
4.2 Powerpoint-lisäosan vaatimukset 13
4.2.1 Valittavat diat 13
4.2.2 Video 14
4.2.3 Esityskonfiguraatio 14
4.2.4 Hylätyt ominaisuudet 14
4.3 Excel 15
4.3.1 Datanrakenne 15
4.3.2 Käytettävyys 16
4.4 Lisäosien jakelu 16
4.4.1 Ympäristöt 16
4.4.2 Rekisteri 17
4.4.3 Arkkitehtuurit 17
5 TOTEUTUS 18
5.1 Addin-rakenne 18
5.2 Yleiset komponentit 18
5.2.1 Ribbon-komponentit 18
5.2.2 JSON-rajapinta 20
5.2.3 Autentikaatio 20
5.2.4 Dokumenttikohtaisten tietojen tallennus 20
5.3 PowerPoint-lisäosa 21

5.3.1 ThisAddin-luokka 21
5.3.2 Ribbon-luokka 22
5.3.3 Image Handler-luokka 22
5.3.4 Video Handler-luokka 22
5.3.5 Tag Handler-luokka 23
5.4 Excel-lisäosa 24
6 POHDINTA 26
LÄHTEET 27

.

ERITYISSANASTO

Active Directory	Windows-ympäristössä toimiva käyttäjä tieto- ja hallintatietokanta.
Autentikointi	Käyttäjän tunnistaminen palvelimen kanssa keskustelua varten.
Custom document properties	Office projektin osa, johon käyttäjä voi tallentaa itsemääritte-lemää dataa.
H.264	MP4-mediaformaatin käyttämä videopakkausstandardi.
Konfiguraatio	Joukko asetuksia, jotka vaikuttavat ohjelman toimintaan.
Lisäosa	Pääohjelmaan ominaisuuksia lisäävä tai muokkaava ohjelma tai ominaisuus.
Ohjelmointiparadigma	Rakenne tai malli, jonka ohjelmointikieli tai tuotettu ohjelma toteuttaa.
Rekisteri	Avain-arvo säilö Windowsissa
Tagi	Tunniste, jolla liitetään järjestelmässä soittolistoja toisiinsa.
UML-kaavio	Kaavio jolla kuvataan ohjelman toimintaa tai rakennetta.
Visualisointi	Asioiden muuttamista visuaaliseksi. Esimerkiksi tuotanto-datan muuttamista graafiksi.
VSTO-lisäosa	Visual studio-ohjelmalla luotu Office-lisäosa.
WMV	Windows media video on Microsoftin kehittämä video for-maatti.

1 JOHDANTO

Opinnäytetyössä toteutettiin Office-tuoteperheen PowerPoint- ja Excel-sovelluksille sisällönjulkaisulisäosat annetun vaatimusmäärittelyn perusteella. Lisäosat toteutettiin VSTO-lisäosina.

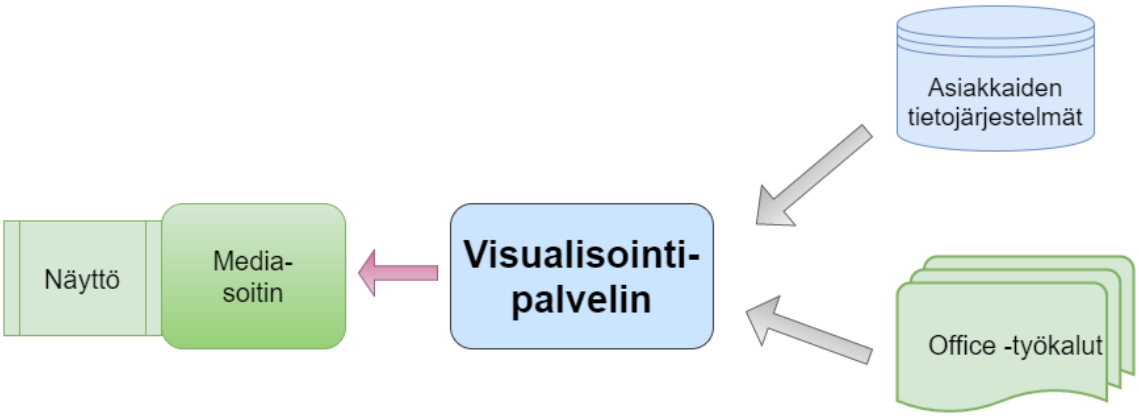
Kokonaisuuden kuvauksessa kuvataan ensin yleisesti ympäristöä ja toimintakokonaisuutta, jossa valmis sovellus toimii. Työn motiivina toimivaa ongelmaa kuvataan ja määritellään yleisellä tasolla kokonaisuuden kuvauksen jälkeen. Määritellyn ongelman perusteella ohjelmille muodostetaan yleiset toiminnallisuusvaatimukset.

Vaatimuksissa määritellään lisäosakokonaisuudelle yksityiskohtaiset vaatimukset. Lisäosat kommunikoivat ulkoisten järjestelmien ja ohjelmien kanssa, näiden ulkoisten komponenttien kanssa kommunikoitiin vaadittavat toteutukset kuvataan tässä luvussa. Työkalut ovat lisäosia olemassa oleville ohjelmille, jotka asettavat kirjasto- ja rakennevaatimuksia työkalujen rakenteelle ja toteutukselle. Luku sisältää myös ohjelmistokohtaiset vaatimukset PowerPointille ja Excelille. Lisäosat asennetaan suoraan käyttäjän tietokoneelle, jolloin asennukselle ja käytölle asettuu toimintaehtoja, jotka tarvitsevat identifioida ja määritellä.

Toteutuksessa esitellään työn teknillinen toteutus. Ensin esitellään tekninen ympäristö, jonka asettaa valmiiksi määritelty lisäosapohja. Lisäosapohjasta esitellään lisäosan kehityksen kannalta tärkeimmät komponentit, sekä lisäosien yhteiset komponentit ja niiden rakenne. Lisäosien rakenne kuvataan UML-kaaviota käyttäen luokkatasolla. Lisäksi luvussa esitellään lisäosien eri toimintojen kulku ja luokkajärjestys.

2 KOKONAISUUDEN KUVAUS

Työssä luodut Office-lisäosat ovat osa visualisointi- ja työnohjausjärjestelmää. Järjestelmä prosessoi tietoa eri datalähteistä. Prosessoidusta tiedosta luodaan visualisointeja, jotka näytetään erilaisilla näytöillä.



KUVA 01. Visualisointijärjestelmän toimintakuvaus

Järjestelmän tarkoituksena on visualisoida yrityksen tai organisaation tietoja, joita voidaan hyödyntää niiden toiminnassa. Tiedot kerätään yrityksen olemassa olevista tietolähteistä. Järjestelmä kytkeytyy erilaisiin tietolähteisiin, kuten tietokantoihin, joista se kerää määritettyä tietoa. Tiedosta prosessoidaan visuaalisia näkymiä, jotka tallennetaan palvelimelle. Mediasoitin hakee tallennetut näkymät palvelimelta ja näyttää visualisoidut tiedot siihen kytketyllä näytöllä. Opinnäytetyössä suunniteltiin ja luotiin Office-lisäosat, joilla täydennetään järjestelmän toimintoja. Tuoden yritysten käytössä olevia Office-tiedostoja voidaan käyttää tietokantana. Tämä helpottaa sisällönjulkaisuprosessia merkittävästi, koska aikaisemmin Office-ympäristössä tuotettu materiaali piti siirtää manuaalisesti palvelimelle.

2.1 Ongelma

Asiakkaiden ja potentiaalisten asiakkaiden yleisimpänä sisällöntuottotyökaluna ovat MS Office-tuoteperheen ohjelmat. Ohjelmien tuottaman materiaalin siirto julkaisujärjestelmään on koettu hankalaksi, epäintuitiiviseksi ja työteliääksi.

Aikaisemmin käytössä on ollut kolmannen osapuolen lisäosa PowerPointille. Käytössä ollut lisäosa helpotti esitysten siirtämistä palvelimelle. Lisäosasta puuttui haluttuja ominaisuuksia, sekä hinta koettiin liian arvokkaaksi, joka johti sen käytön lopettamiseen.

Excel-ohjelmalle ei aikaisemmin ollut käytössä lisäosaa, jolla tiedot saataisiin siirrettyä helposti suoraan Excel-dokumentista palvelimelle.

Sopivaa valmista ohjelmistoa on ollut haasteellista löytää, sillä tietojen siirto on pakotettu lisäosien määrittelemään tiettyyn formaattiin olemassa olevissa lisäosissa, eikä järjestelmään haluttu tehdä rajapintaa yksittäistä tapausta varten.

2.2 Työn tehtävä

Työn tehtävänä on etsiä ratkaisu, jolla Office-ohjelmat saadaan integroitua visualisointijärjestelmään. Ratkaisuksi valittiin lisäosien luominen itse ulkopuolisen ratkaisun sijasta.

Työssä määritellään, suunnitellaan ja tuotetaan työkalut, joiden avulla parannetaan julkaisujärjestelmän yleistä käyttökokemusta helpottamalla sisällön siirtämistä ja päivittämistä järjestelmään datalähteestä, joka on asiakkaiden yleisessä käytössä.

2.2.1 PowerPoint-lisäosa

PowerPoint lisäosan tarkoituksena on helpottaa luotujen esitysten siirtämistä järjestelmään. Lisäosan tehtävänä on automatisoida esitysten siirto ja muokkaus, niin ettei käyttäjien tarvitse käyttää muita työkaluja esityksen päivittämiseksi tai julkaisemiseksi. Lisäosa muuttaa sisällön esitysformaattia PowerPointin omasta formaatista yleisesti käytettävämmäksi. Esitys muutettaan kuviksi, tai MP4-videoksi, joita voidaan käyttää helposti visualisointijärjestelmässä.

2.2.2 Excel-lisäosa

Monien järjestelmän käyttäjien toiminnoista ja prosesseista tuotettava data on Excel pohjaista taulukkodataa. Dataa joko hallinnoidaan suoraan tai sitä koostetaan Excelissä. Excel-lisäosan tehtävänä on helpottaa Excel-taulukkodatan siirtämistä visualisointijärjestelmään. Lisäosa automatisoi siirtoa erilaisten formaattimuokkausten, valinta- ja muutostyökalujen avulla, sekä helpottaa datan päivitysprosessia automatisoidun päivityksen muodossa.

3 TYÖKALUT JA KIELET

3.1 Visual Basic

Visual Basic (VB) on Microsoft oy:n Windows-alustoille luoma ohjelmointikieli. Microsoft tarvitsi 90-luvun alussa kielen, jolla voitiin luoda nopeasti yleistyviä graafisia käyttöliittymiä helposti. Microsoft halusi luoda kielen, jolla oli helppo luoda graafisia käyttöliittymiä, joiden luonti oli aikaisemmin huomattavan työlästä. (Mack , 2002). Uuden kielen pohjaksi valittiin John Kemenyn ja Tom Kurtzasin vuonna 1962 Darthmouthin yliopistossa kehittämä Basic. Basic suunniteltiin yliopiston käyttöön ohjelmoinnin opettamisen helpottamiseksi. Käyttötarkoituksensa takia kielen ominaisuudet suunniteltiin mahdollisimman helpoiksi oppia, sekä oppimiskäyrästä matala. (Bellis, 2015).

Visual Basic noudattaa ohjelmointiparadigmaltaan olio-ohjelmointia. Paradigmassa ryhmä erilaisia toisistaan riippuvia toimintoja ja arvoja on enkapsuloitu luokan sisään (Microsoft: Object-Oriented Programming. 2016). Ohjelman etenemisessä kieli noudattaa tapahtumajohteista paradigmaa, jossa ohjelman suorituksen eteneminen on sidottu tapahtumiin, jotka esimerkiksi voivat käynnistyä eri käyttöliittymäkomponenteista. Tapahtumapohjainen suoritusmalli keskittyy vastaamaan eri suorituksen aikana syntyneisiin tapahtumiin. Tapahtuman suorituksessa ei ohjelma ota kantaa ohjelman aikaisempaan tilaan, vaan käsittelee tapahtumaa itsenäisenä komponenttina. (TechnologyUK, 2016).

```
For i = 0 To pl.tags.Count - 1
    For j = 0 To tags.Count - 1
        If pl.tags.Item(i).tag = tags.Item(j).tag Then
            Tag.SelectedItems.Add(tags.Item(j))
        End If
    Next
Next
```

KUVA 02. Kaksiulotteinen for silmukka if-ehtolauseella

3.2 Visual Basicin erityispiirteitä

Visual Basicin syntaksi on lähellä kirjoitettua kieltä (kuva 01). Kieli käyttää erilaisten loogisten operaatioiden esittämisessä kirjallista muotoa symboleiden sijaan. Esimerkkisi operaattori: ja C++ kielessä esitettään symboleilla &&, joka esitetään muodossa And. Kielessä ei erotella suoritettavia lauseita keskenään millään erillisellä lopetusmerkillä,

vaan kääntäjä päättelee lauseiden päättymisen rivien mukaan. Visual Basic ei erottele pieniä ja suuria kirjaimia, vaan se käsittelee isot ja pienet kirjaimet syntaksisesti samaksi. (Microsoft: Visual Basic Language Reference. 2016).

3.3 Visual Studio Tools for Office -kehitystyökalut

Visual Studio Tools for Office -kehitystyökalut (VSTO) on Visual Studio –kehitysympäristöön tarkoitettu lisäosa. VSTO:lla tuotetaan Office -ohjelmistoon lisäosia. Kehitystyökalut sisältävät lisäosia varten tarvittavat projektipohjat ja käytönaikaisen ajoympäristön.

Projektipohja sisältää tarvittavat komponentit lisäosan suorittamista varten. Pohja sisältää konfiguraatio -tiedostot, joiden avulla Office-ohjelmat suorittavat lisäosat. Pohjan mukana tulee luokka, joka suoritetaan ensimmäisenä lisäosan käynnistyessä. Visual studio -kehitysympäristö tukee pohjaa tarjoamalla erilaisia työkaluja kehitykseen, kuten työkalut tekstien kielilokalisoimiseen. (Microsoft: Programming VSTO Add-Ins. 2017).

Office -ohjelmat ajavat lisäosia niihin liitetyn VSTO runtime -kirjaston avulla. Ajoympäristö koostuu komponenteista, joiden avulla lisäosa kommunikoi ohjelman eri toimintojen kanssa. Ajoympäristön kirjastojen kautta lisäosa voi lukea Office-ohjelmien dokumenttien tietoja, sekä muokata niitä.

4 VAATIMUKSET

4.1 Yleiset vaatimukset

4.1.1 Versiotuki

Lisäosille määritettiin vähimmäisversioksi Office 2010-versio ja uusimmaksi tuetuksi versioksi Office 2016. Tuettujen versioiden tuli kattaa suurin osa yritysten käyttämistä versioista. Asiakkaiden käytetyimpänä versiona oli 2010-versio, joka määritti pääkehitysversion. Ominaisuudet ja tekniset ratkaisut suunniteltiin käytetyimmälle versiolle. Pienintä tuettua versiota kartoitettaessa 2010 vuoden versiota aikaisemmat versiot karsiutuivat pois niiden pienen markkinaosuuden takia, sekä vanhemmista versioista puuttuvien ohjelmistorajapintojen takia. Aiemmat versiot eivät sisällä samanlaista lisäosarajapintaa, kuin 2010-versio ja sitä uudemmat versiot. Vanhojen versioiden tuki todettiin kannattamattomaksi niiden erilaisen rakenteen tuottaman lisätyön takia.

4.1.2 Autentikointi ja autorisointi

Lisäosien käyttäjähallinan vaatimuksena oli kaikkien lisäosien ja palvelimen välisen kommunikoinnin tapahtuvan autenttikoidusti. Kaikki kutsut tuli olla yksilöitävissä käyttäjään. Autentikaatiossa tuli pystyä sisällönhallintajärjestelmän käyttäjätunnusia, joiden hallinta tapahtuu keskitetystä käyttäjähallinnasta. Lisäosien toiminnallisuutta tai jakelua ei rajoitettu sulkemalla toimintoja autentikaation taakse. Käyttäjän tunnukset ovat yhteiset koko järjestelmässä, jolloin ei synny ylimääräistä ylläpitokuormaa useiden eri tunnuksien hallinnoinnista.

4.1.3 HTTP-protokolla

Asiakkaille tehtyjen asennusten perusteella oli tarve tukea erilaisia verkkokonfiguraatioita ja verkkovaatimuksia. Perustoiminnalla lisäosien kommunikointi palvelimen kanssa tapahtuu salatun yhteyden välityksellä. Osa asiakkaiden sisäverkossa tapahtuva liikennöinti tapahtuu salaamattomana. Näihin sisäverkkoasennuksiin tulee lisäosien tukea myös salaamatonta yhteyttä.

4.1.4 Tilan säilyttäminen

Lisäosien tärkein tehtävä on helpottaa ja suoraviivaistaa käyttäjän käyttökokemusta. Lisäosat sisältävät erilaisia konfiguraatio- ja tilatietoja, joiden uudelleen asettaminen on työlästä ja aikaa vievää. Lisäosan asetustiedot tallennetaan dokumenttiin, jolloin ne eivät ole järjestelmäriippuvaisia. Lisäosissa erilaisten asetusten muistamisen ja automatisoinnin avulla saavutettiin parempi käyttäjäkokemus. Asiakkaiden ei tarvitse tehdä muutoksia ja päivityksiä järjestelmässä useaan paikkaan. Lisäosa helpottaa esimerkiksi näytettävän sisällön päivitystä järjestelmään. Sisällön päivitys tapahtuu yhdellä painalluksella dokumentissa, josta näytettävän esityksen materiaali haetaan.

4.2 Powerpoint-lisäosan vaatimukset

Powerpoint lisäosan tehtävänä on muuntaa esitys, tai sen osa muotoon, jossa se voidaan siirtää ja näyttää palvelimella ja mahdollisimman monella erilaisella päätelaitteella. Powerpointin tärkein tehtävä on helpottaa visuaalisen sisällön julkaisua ja keventää vaiheketjua, joka vaaditaan sisällön julkaisemiseen. Powerpoint on yleisimpiä käytettyjä esitysohjelmia, joten sen käyttäminen on luontaista käyttäjille. Käyttäjä valitsee, mitkä materiaalit esityksestä julkaistaan. Julkaistavalle esitykselle valitaan tunnuslistasta kohteet, jolle haluttu esitys linkitetään. Julkaistu esitys linkittyy automaattisesti järjestelmässä haluttuihin ryhmiin.

4.2.1 Valittavat diat

Diaesityksestä halutaan näyttää osa esityksestä tai koko esitys. Käyttäjä valitsee, valitaanko koko esityksen sisältö julkaistavaksi julkaisujärjestelmässä, vai valitaanko osa dioista lähetettäväksi. Lisäosa säilyttää osan esityksen esitysmäärityksistä, kuten esitykseen määritellyt diojen vaihtoajat. Palvelin ei kykene ottamaan PowerPoint-esitystä tai sen osia suoraan. PPTX-formaatin monimutkaisuuden vuoksi palvelimelle ei ole toteutettu sen automaattista käsittelyä. Esitys tai sen osat muunnetaan kuviksi esitysformaatista, jotka lisäosa lähettää palvelimelle.

4.2.2 Video

Esitysten ajastukset voidaan säilyttää kuvien lähetyksessä, mutta esitykseen määritettyjen animaatioiden ja efektien säilyttäminen kuvamuotoon muuntamisessa on mahdotonta. Esityksen animaatioiden ja tehosteiden säilyttämiseksi PowerPoint mahdollistaa esityksen muuttamisen videoksi. Video säilyttää kaikki esitykselle annetut animaatiot ja tehosteet.

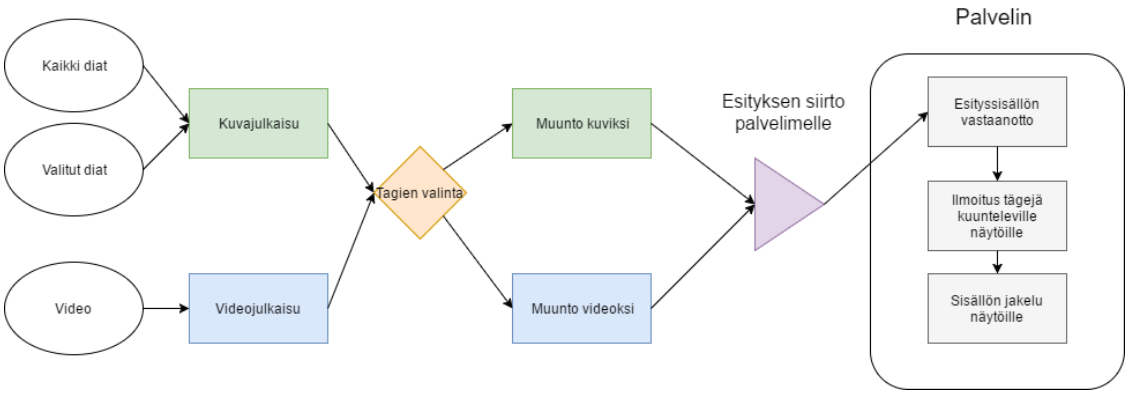
Palvelin hyväksyy vain mp4-tyyppistä videoformaattia, jota kaikki Office-versiot eivät kykene tuottamaan. Office 2010-versiossa esityksestä voidaan muodostaa vain wmv-formaatin videota. Palvelimen kanssa yhteensopimaton video muunnetaan mp4-muotoon käyttäjän tietokoneella ennen palvelimelle lähetystä. Lisäosa tuottaa wmv-tyyppistä videota, joka ohjataan erilliseen videomuuntimeen, jossa video muunnetaan palvelimelle sopivaan mp4-formaattiin. Muunnettu video lähetetään julkaistavaksi palvelimelle.

4.2.3 Esityskonfiguraatio

Lähetettävä media liitetään palvelimella esityslistaan. Esityslista sisältää ladatun sisällön lisäksi ajastuksia ja muita esitykseen ja sen kulkuun liittyviä tietoja. Lisäosan tuli hallita uuden esityslistan luonti lähetettävälle materiaalille ilman muita toimenpiteitä. Esitystä päivitettäessä uudella materiaalilla lisäosan tuli hallita materiaalin päivitys oikeaan esityslistaan ja säilyttää esityslistan asetukset. Esityslista kytketään muuhun esitettävään materiaaliin tageilla. Esityksen julkaisussa hallitaan lisäosassa esityksen lähetyksen yhteydessä, jolloin esitykselle valitaan palvelimelta ladatut tagit, johon esitys kytketään.

4.2.4 Hylätyt ominaisuudet

Osa julkaisun automatisointia PowerPointin-lisäosassa oli automaattinen esityksen päivittäminen palvelimelle tallennuksen yhteydessä. Ominaisuudesta valmistettu prototyyppi osoittautui epäkäytännölliseksi. Materiaalin siirto palvelimelle julkaisukoneelta vei ajallisesti huomattavan ajan. Lähetysprosessi lukitsi käyttäjän ohjelmiston lähetyksen ajaksi, joka teki lisäosan käyttökokemuksesta huonon.



KUVA 03. PowerPoint-esityksen julkaisuprosessi

4.3 Excel

Excel lisäosan tehtävänä on siirtää taulukkomuotoista dataa palvelimelle. Tyypillinen siirrettävä data on prosessi tai tilannedataa, jonka rakenne ja muoto vaihtelevat datalähteittäin. Excelistä palvelimelle siirretty data käsitellään ja siitä muodostetaan erilaisia visualisointeja, jotka julkaistaan haluttujen näyttöjen esityksissä.

4.3.1 Datanrakenne

Visualisointipalvelimen rajapintatoteutus määrittelee Excelistä kerättävän datan lähetys muodon, eli formaatin. Datassa määritellään riveille tai sarakkeille avainarvot, joiden lisäksi datalle voidaan määrittää aikasarja, jolla dataa voidaan ryhmitellä palvelimella.

Asiakkaiden käyttämistä Excel-pohjista on analysoitu datan peruskäyttötapoja, joista on koostettu minkälaisia määrittelyvaihtoehtoja ja rajauksia datalle tulee voida asettaa. Kerättävän datan sarakkeiden ja rivien määrä voi vaihdella dokumentissa. Lisäosa tunnistaa dynaamisesti ennalta määritellyltä riviltä avainten ja aikasarjojen alku- ja loppusolut. Alku- ja loppusolujen perusteella lisäosa kerää datan ja liittää sen määriteltyihin avain- ja aika-arvoihin. Tiiviissä Excel dokumentissa voi olla useita toisistaan riippumattomia taulukoita, jolloin automaattinen datan rajojen tunnistus ei toimi. Ongelman ratkaisemiseksi datan äärirajoja on mahdollista määritellä pienemmäksi, kuin datan äärirajat.

Excel antaa mahdollisuuden erilaisten omien datatyyppien luontiin. Järjestelmän palvelin ei kykene tunnistamaan kaikkia itse luotuja tai maalokalisoituja datatyyppejä. Lisäosa

kykenee tekemään muunnoksia yleisimmille datatyypeille; näitä ovat esimerkiksi päiväformaatit, jotka muunnetaan unix-aikaformaattiin, sekä desimaalipilkun muuntaminen pistemuotoon.

4.3.2 Käytettävyys

Asiakaspohjista todettiin, ettei termejä, joita halutaan käyttää eivät vastaa dataa visualisoitaessa. Käyttäjän tulee kyetä muuttamaan palvelimelle lähetettävän datakokonaisuuden avainarvot, niin ettei se tee muutoksia alkuperäiseen taulukkoon.
Yhdessä Excel dokumentissa voi olla kerättävää dataa usealla eri välilehdellä ja jokaisella eri välilehdellä on muotomääritykset datalle. Käyttökokemuksen kannalta on olennaista, että jokaiselle datalähteelle asetetut asetukset säilyvät dokumentissa. Dokumentille on mahdollista asettaa automaattinen tietojen lähetys tallennuksen yhteydessä. Lehdille, joille on asetettu automaattinen lähetys. Lisäosa lähettää muuttuneen datan tallennettujen määritysten mukaisesti.

4.4 Lisäosien jakelu

Lisäosien jakelu tulee olla mahdollisimman helppoa. Asiakkaiden tulee kyetä asentamaan Office-lisäosat autonomisesti ilman erillistä ohjausta ja ylimääräisiä toimenpiteitä. Jakelun tulee olla yhteensopiva isojen yritysympäristöjen kanssa.

4.4.1 Ympäristöt

Yritysten tietohallinto on usein keskitettyä ja työasemat ovat tarkasti hallittua. Lisäosien asennuspaketit noudattavat hallinnoidun Windows-ympäristön yleisiä toimintatapoja ja vaatimuksia. Asennuspaketit on mahdollista asentaa etänä tietokoneille. Asennuksessa asennuspaketti on mahdollista suorittaa komentoriviltä äänettömässä tilassa, jossa asennus ei indikoi asennettavan laitteen käyttäjälle prosessista, vaan se suoritetaan taustaajona, kuten keskitetyssä hallinnassa on tapana.

4.4.2 Rekisteri

Asennuksen aikana tietokoneelle asetetaan useita eri Windows-rekisteriavaimia. Lisäosat vaativat useita eri avaimia toimintaan. Osa avaimista määrittää lisäosan käyttäytymistä sovelluksessa, kuten lisäosan tyypin ja käynnistystiedot, sekä lisäosan asennuspaikan. Lisäosan toimintaan vaikuttavat rekisteriavaimet asennetaan koneen yleiseen rekisteriavainkantaan (local machine). Avaimia ei voida asettaa käyttäjäkohtaisiksi, koska lisäosan asennus tapahtuu erillisellä sovellushallintaan tarkoitetulla käyttäjällä Active Directory-ympäristössä. Lisäosien käytönaikaiset rekisteriavaimet tallennetaan käyttäjäkohtaisesti käyttäjän omaan rekisteriin, johon ei ole asetettu rajoituksia hallintaympäristöstä.

4.4.3 Arkkitehtuurit

Sovellus tukee sekä 32- ja 64-bittisiä Windows-käyttöjärjestelmä variantteja . 32-bittinen ympäristö ei ole asiakkailla yleinen, mutta lisäosille asetettu mahdollisimman laaja käyttöympäristötavoite vaatii vähemmän käytetyn arkkitehtuurin tukemista. Lisäosien sisäisen toiminta on riippumaton käyttöjärjestelmän bittisyydestä. Käytetty arkkitehtuuri on ohjelman kääntämistä varten, sekä lisäosaa asennettaessa, jolloin määritellään rekisteriavainten sijainti. Rekisterin rakenne eroaa toisistaan eribittisissä Windows-versioissa. Office-tuoteperheestä on olemassa molemmat variantit, joiden rekisteriavaimet eroavat keskenään. 64-bittisen lisäosan tulee huomioida molemmat Office-ohjelmistoversiot.

5 TOTEUTUS

5.1 Addin-rakenne

VSTO-addin koostuu joukosta VSTO-spesifikaation mukaisia valmiskomponentteja ja varsinaisista tässä työssä toteutetusta ohjelmakoodista. Ennalta määritellyt komponentit koostuvat rekisteriavaimista, manifest-konfiguraatiosta ja Office Runtime- kirjastoista. Rekisteriavaimet sisältävät yleistietoja komponentin nimestä, tekijästä ja ohjaa ohjelmistoa lisäosan käytöstä. Manifest-konfiguraatiotiedostot sisältävät lisäosan asennuksessa ja toiminnassa käytettyjä asetuksia. Lisäosan asennuksessa käytettävä deployment manifest määrittelee lisäosalle asennuspaikan mahdolliset pohjavaatimukset komponenttien toiminnalle, sekä tiedot lisäosan mahdollisista versiopäivityksistä. Application manifest sisältää tiedot lisäosan versiosta, ohjauksen lisäosan käynnistyksestä, sekä lisäosan käynnistyksessä aloitusluokan. Manifest myös määrittelee osan lisäosan ulospäin suuntautuvasta kommunikoinnista, kuten logiasetuksista. (Architecture of VSTO Add-ins, 2016). Valmiiden komponenttien merkittävin osa koostuu joukosta Visual Basic-kirjastoja, jotka sisältävät kirjastoja käyttöliittymäkomponenteista tiedostojen käsittelyyn.

5.2 Yleiset komponentit

Kaikille lisäosille on määritelty pääluokka, jota ohjelma kutsuu lisäosaa käynnistettäessä. Lisäosissa luokan tehtävänä on kuunnella ja hallita erilaisia ohjelmassa tapahtuvia tapahtumia, joista se välittää tiedon muille lisäosan komponenteille. Pääluokka hallitsee myös dokumenttikohtaisten konfiguraatioiden luonnin, tallennuksen ja lukemisen.

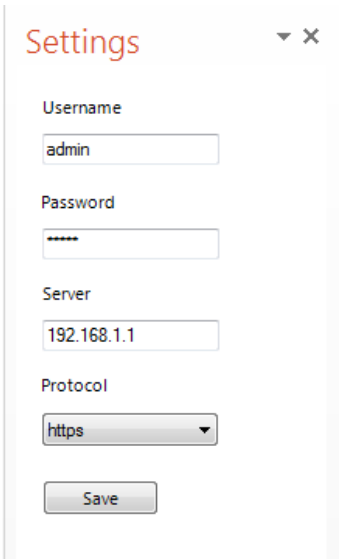
5.2.1 Ribbon-komponentit

Lisäosien tavoitteena on toimia mahdollisimman luonnollisena lisänä Office-ohjelmistoille. Tämä tarkoittaa erilaisten toimintatapojen ja rakenteiden käyttöä. Ohjelmistoissa dokumenteille tehtävät operaatiot on jaettu eri välilehtien alle. Välilehti sisältää joukon toimintoja ja työkaluja, jotka kohdistuvat johonkin dokumentin osa-alueeseen tai sen muotoon. Välilehti voi sisältää sisäistä jaottelua, jossa komponentit ja työkalut ovat jaoteltu omiksi ryhmiksi niiden toiminnan mukaan.

KUVA 04. PowerPoint lisäosan ribbon-välilehti

Lisäosien välilehdet noudattavat kuvan 02 mukaista muotoa. Kaikki lisäosan toiminnot on koottu yhden välilehden alle, joka on nimetty yrityksen mukaan. Välilehdellä toiminnot on ryhmitelty eri toiminnollisuuksien mukaan, kuten PowerPointin tapauksessa, jossa eri julkaisuvaihtoehdot on koottu omaksi ryhmäksi ja lisäosan asetukset erotettu omaksi ryhmäkseen.

Lehdet sisältävät vain yksinkertaisia asetuksia tai nappeja, joita painamalla käyttäjä pääsee kattavampiin ominaisuuksiin tai asetuksiin. Näiden toimintojen näyttämiseen käytetään Officessa kahta tapaa. Alkuperäinen vanhempi tapa on avata erillinen ikkuna, joka sisältää avatun komponentin sisällön. Uudempi tapa on avata ohjelmistoikkunassa lehti, joka sisältää samat tiedot. Lehdet aukeavat ohjelman oikealle laidalle kaventaen työskentely aluetta. Lisäosien toiminta tapahtuu aina dokumenttien muokkausten välissä, johon selkeä huomiota herättävä komponentti soveltuu paremmin.



KUVA 05. Lisäosien asetuslehti

5.2.2 JSON-rajapinta

Palvelin tarjoaa REST-tyyppisen ohjelmointi rajapinnan. Visual Basicin standardikirjasto sisältää HTTP-protokollatyökalut, mutta ne ovat rakenteeltaan alkeellisia, joten ne eivät sovellu suoraan REST-pohjaiseen keskusteluun. REST-pinnalle luotiin luokka, jonka tehtävänä on lähettää ja vastaanottaa JSON-dataa. Luokan metodit saavat parametrina palvelimen tarvitsemia tietoja, kuten tarkan osoitteen ja kirjautumistiedot. Lähetettäessä metodi saa lisäksi parametrina JSON-muotoista dataa, jonka lisäosa lähettää palvelimelle. Metodi palauttaa vastauksena onnistuiko lähetys palvelimelle ja hyväksyikö palvelin JSON:in.

5.2.3 Autentikaatio

Tietoturvan ylläpitämiseksi palvelin hyväksyy vain autentikoituneen kommunikaation. Palvelimen kommunikaatiorajapinta hyväksyy kutsut, jotka ovat signeerattu käyttäjän toimesta, sekä hylkää kutsut jotka tulevat käyttäjältä, jolla ei ole oikeuksia rajapintaan.

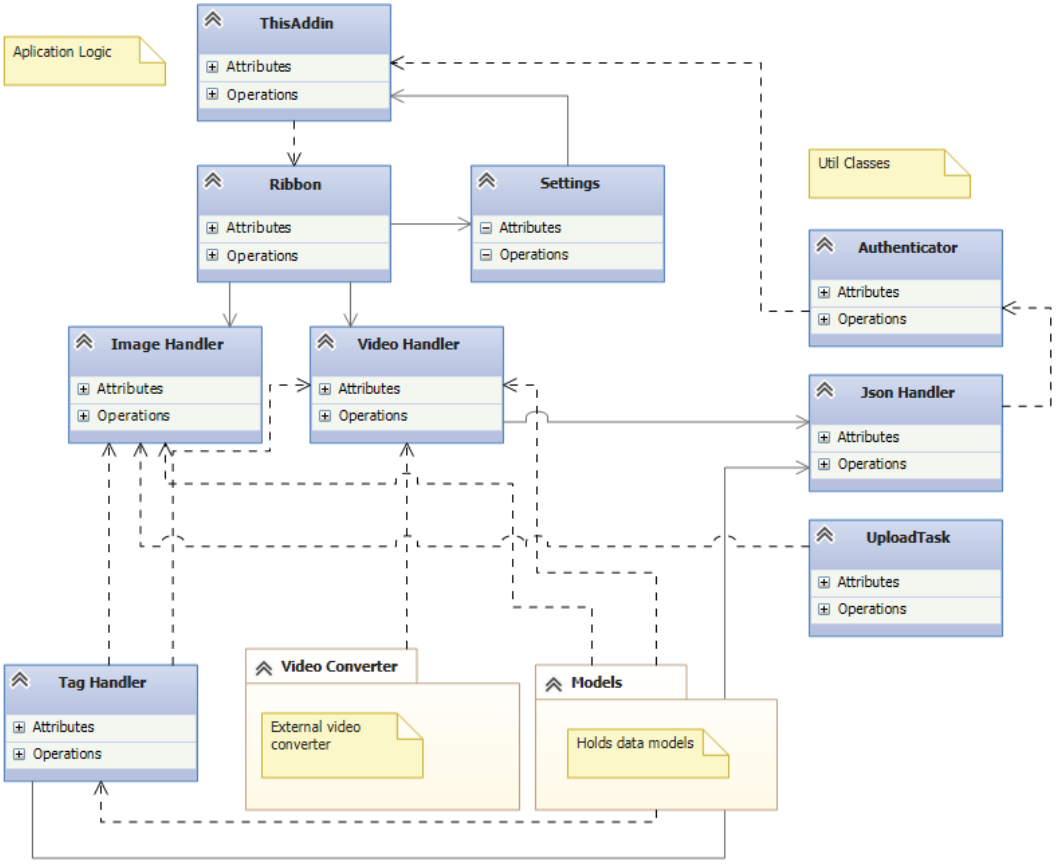
Lisäosan toiminta poikkeaa muista palvelimen kanssa kommunikoivista komponenteista, johtuen VS:n HTTP-protokollakirjaston rajoitteisuudesta, vaatii lisäosa erillisen helpotetun autentikaatiometodin toteuttavan rajapinnan. Muiden komponenttien kommunikaatiosekvenssi eroaa lisäosista, joten lisäosille täytyi luoda oma rajapinta. Autentikaatiorajapinta noudattaa samaa rajapintarakennetta ja kommunikaatiota kuin muut palvelimen tarjoamat informaatiorajapinnat. Autentikaatiorajapinta lähettää ja vastaanottaa JSON-muotoista dataa. Rajapinta vastaanottaa kirjautumiseen vaadittavat tiedot ja palauttaa palvelimen rajapintojen vaatiman autentikaatiotiedon. Lisäosissa autentikoitumisesta vastaa erillinen luokka, joka tukeutuu json-luokkaan palvelimen kanssa kommunikoinnissa. Luokka vastaanottaa käyttäjän kirjautumistiedot ja palauttaa palvelimen vaatimat tunnistetiedot. Luokka ei pidä sisällään tunnuksia tai tunnistetietoja, vaan rakentaa vaaditun kommunikaation autentikoitumiseen sen saamista parametreista ja välittää saadut tunnistetiedot eteenpäin.

5.2.4 Dokumenttikohtaisten tietojen tallennus

Lisäosien käyttökokemuksen parantamiseksi lisäosat tallentavat tietoja uudelleenkäyttöä varten. Tietojen paikkansapitävyyden ylläpitämiseksi poissuljettiin ratkaisut, joissa tiedot

tallennetaan dokumentin ulkopuolelle. Tietojen tallentamiseen valittiin ensimmäisenä custom document properties-muuttujat. Tallentamisratkaisu hylättiin, kun todettiin tallennettavien tietojen koon olevan suurempi kuin custom document propertiesille-määritelty enimmäismerkkiraja 255 merkkiä. (Microsoft CustomDocumentProperties, 2017) Korvaavaksi ratkaisuksi valittiin custom XML-tekniikka (Microsoft custom XML, 2017). Lisäosat tallentavat asetukset dokumentin tallennuksen yhteydessä ja lukevat asetukset dokumentin avauksessa.

5.3 PowerPoint-lisäosa



KUVA 06. UML-luokkakaavio PowerPoint lisäosasta

5.3.1 ThisAddin-luokka

Lisäosan juuriluokkana on ThisAddin-luokka, joka on valmiina lisäosa projektin aloitus konfiguraatiossa. Luokkaa voidaan kutsua mistä tahansa lisäosaa, sillä se on määritelty lisäosan globaalissa nimiavaruudessa. PowerPoint lisäosan ThisAddin-luokka sisältää dokumentin elinkaareen kuuluvien tapahtumien seurannan, kuten käsiteltävien doku-

menttien esityksen avaus ja sulku tapahtumia. Avaus tapahtumassa luokka hakee dokumentin sisäisistä tiedostoista lisäosan asetusluokan (aikaisemmin määritelty custom XML). Sulkutapahtumassa luokka tallentaa asetukset dokumentin sisään uudelleen käyttöä varten.

ThisAddin-luokka sisältää Authenticator-luokan, joka ylläpitää kirjautumistietoja, ja hallinnoi niiden vanhenemista ja uudelleen hakemista. Autentikaatioluokkaa kutsutaan ThisAddin luokan kautta. Luokka palauttaa palvelinrajapintakommunikoinnissa vaadittavan tokenin. Se ylläpitää sisäisesti tokenin voimassaoloaikaa, sekä hakee uuden tokenin vanhan vanhentuessa. Näin muiden lisäosien luokkien ei tarvitse ottaa kantaa autentikoitumisen ylläpitoon.

5.3.2 Ribbon-luokka

Ribbon-välilehti sisältää painikkeet lisäosan eri toimintoihin. Se ylläpitää avonaisena olevaa lehteä ja tarjoaa muille luokille rajapinnan lehtien avaamiseen ja sulkemiseen.
Asetusvälilehti pitää sisällään lisäosan asetusten muokkaamiseen tarvittavat komponentit, jotka se hakee ThisAddin-luokalta ja päivittää ne asetusten tallennuksessa.

5.3.3 Image Handler-luokka

Image Handler vastaa koko esityksen tai valittujen diojen lähettämisestä. Luokka käyttää Tag Handleriä liittämään lähetettävän esityksen palvelimella oleviin soittolistoihin. Lisäosan testauksessa todettiin esityksen lähettämisen palvelimelle kestävän huomattavan kauan, jolloin se vaikuttaa negatiivisesti käyttäjäkokemukseen. Lisäosa nopeuttaa esityksen lähettämistä palvelimelle pakkaamalla esityksen kuvat ZIP-pakettiin. Pakatut kuvat poistetaan lähetyksen päätyttyä. Pakettien lähetyksestä vastaa UploadTask niminen luokka, joka asynkronisesti lähettää tiedostot palvelimelle. UploadTask pitää sisällään .NET kirjaston WebClient luokan, joka vastaa http-kommunikaatiosta palvelimen kanssa (Microsoft WebClient Class, 2017).

5.3.4 Video Handler-luokka

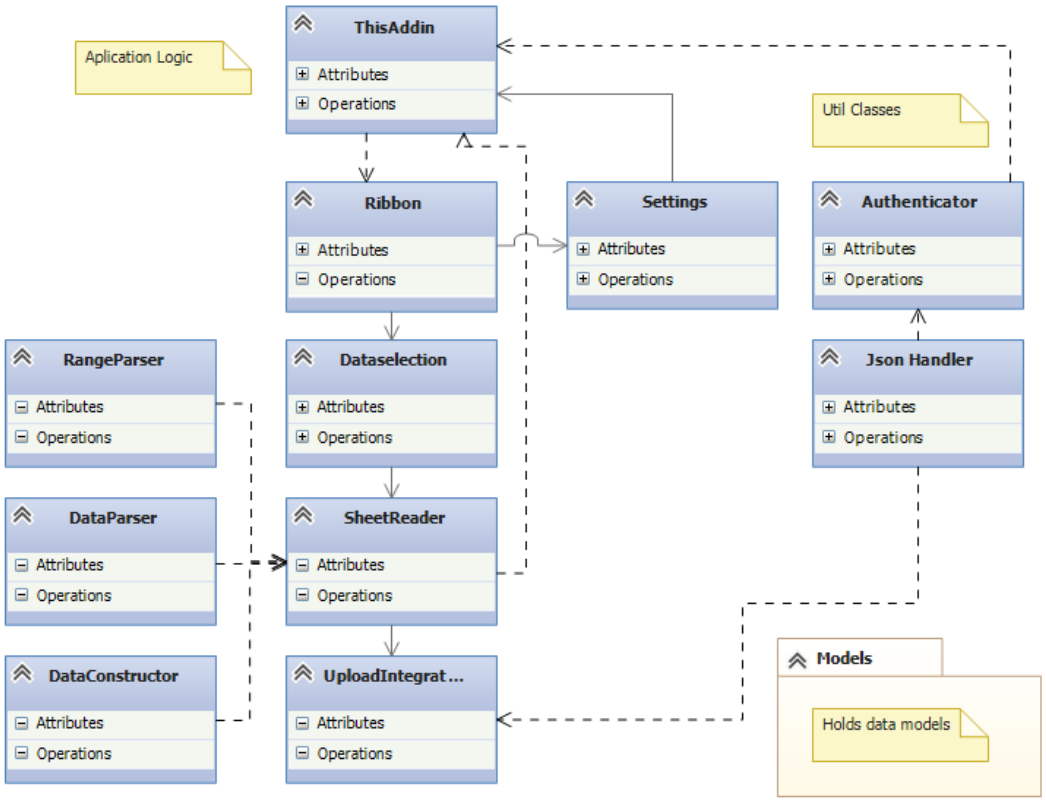
Video Handler vastaa esityksen lähettämisestä videona. PowerPointin video muodostus ei kykene kaikilla versioilla tuottamaan haluttua MP4-videoformaattia. MP4-tuki on

käytössä 2013-versiosta lähtien. (Office Support, 2017). Lisäosan version vähittäisvaatimuksesta (Office 2010) johtuen joudutaan video muodostamaan ensin WMV muotoon. Video Handler kutsuu erillistä videomuunnin ohjelmaa, joka muuntaa muodostetusta videosta palvelimen kanssa yhteensopivan videon. Videon lähetyksestä vastaa UploadTask-luokka. Lähetetty esitys liitetään palvelimen soittolistoihin, kun video on siirretty palvalimelle käyttäen TagHandler-luokkaa.

5.3.5 Tag Handler-luokka

TagHandler hakee palvelimelta listan kuvauksista (TAG), joita käytetään esitysten ryhmittelyssä. Tagit avataan omalle lehdelle, josta käyttäjä valitsee mihin tageihin esitys liitetään. TagHandler lähettää valitut tagit palvelimelle, joka kytkee esityksen tageihin. PowerPoint lisäosan kaikki data on erotettu omiin dataluokkiin. Luokat noudattavat Scalan case class-tyyppistä ajattelua, missä luokka sisältää muuttujia ja normaalista olio-ohjelmoinnista poiketen ei sisällä logiikkaa luokan muuttujien käsittelyyn (Scala case class, 2017). Model-luokat ovat erotettuna omaksi kokonaisuudeksi, jolloin ne ovat selkeästi erotettuna ohjelmistologiikasta.

5.4 Excel-lisäosa



KUVA 07. UML-luokkakaavio Excel lisäosasta

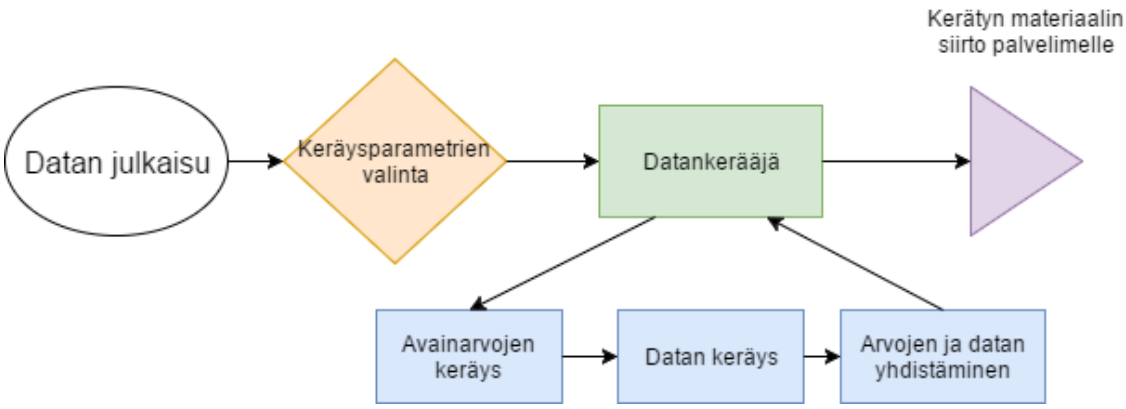
Excel lisäosan ThisAddin-luokka sisältää samat toiminnot kuin PowerPointin vastaava luokka. Luokka kuuntelee lisäksi dokumentin tallennus tapahtumia, jos lisäosan asetuksissa on määriteltynä automaattinen datan lähetys tallennuksen yhteydessä käynnistää ThisAddin datan keräys ja lähetysketjun.

Settings-luokka on identtinen vastaavan PowerPoint-luokan kanssa.

Ribbon-luokka sisältää painikkeet asetus ja datan valintalehdille. Se vastaa lehtien ylläpidosta ja lehtien käyttämien resurssien varaamisesta ja vapauttamisesta.

Dataselection-luokka vastaa datamallin määrittelemisestä ja luonnista. Luokka sisältää lehden, jossa käyttäjä määrittelee Excel dokumentin datalukumallin aktiiviselle välilehdelle. Luokka lukee tallennetun datamallin ThisAddin-luokasta ja tallentaa muokatun datamallin takaisin ThisAddin-luokassa sijaitsevaan konfiguraatioon. Käyttäjän määriteltyä datamallin, luokka tarkastaa mallin oikeellisuuden ja hyväksytty malli annetaan SheetReader-luokalle.

SheetReader-luokka vastaa datan keräämisestä Dataselection-luokalta saadun datamallin mukaisesti. Se käyttää RangeParser-, DataParser- ja DataConstructor-luokkia datan keräämisessä ja muokkaamisessa. SheetReader etsii dokumentista datamallin määrittelemän välilehden, josta data luetaan. Saadusta datamallista luokka irrottaa avain- ja aikamääritykset, jotka lähetetään RangeParser-luokalle määritellyn välilehden kanssa. RangeParser analysoi saadusta avain- ja aikamäärittelystä näiden orientaatiot ja paikat saadulla välilehdellä. Välilehdeltä luetaan määrittelyn mukaiset alueet, jotka kerätään ja palautetaan SheetReaderille. Avainarvojen keräyksen jälkeen SheetReader lähettää datamallista arvomääritykset DataParserille. DataParser lukee välilehdeltä arvot, jotka se seuloo datamallissa olevien rajoituksen mukaisesti. Arvoja lukiessa Parser muuntaa osan arvojen tyypistä palvelimen käyttämään muotoon. Muunnos säännöt ja muunnettavat arvot luokka lukee Models-paketista sijaitsevista säännöistä. Arvojen keräyksen ja muuntamisen jälkeen ne palautetaan SheetReader-luokalle. Avainten ja arvojen keräyksen jälkeen kerätyt arvot annetaan DataConstructorille. Constructor muodostaa avaimista ja arvoista palvelimen käyttämän datamallin. Muodostettu datamalli ohjataan UploadIntegratio-luokalle. Luokka sarjottaa mallin jsoniksi ja lähettää sarjotetun datamallin palvelimelle eri kohteisiin, käyttäen JsonHandler-luokkaa.



KUVA 08. Excel-datan julkaisuprosessi

6 POHDINTA

Opinnäytetyön tarkoituksena oli tuottaa työkalut, joilla parannettiin näytetyön tilaajan visualisointituotteen käytettävyyttä. Näytetyö palautti työn tilaajan tuotteesta poistuneita ominaisuuksia ja paranteli niitä, sekä loi uusia tarpeelliseksi koettuja ominaisuuksia. Työn alkaessa oli valmiiksi kartoitettuna, että työssä tehdyt lisäosat oli teknisesti mahdollista toteuttaa. Projektin tuloksena oli Office-ohjelmille lisäosat, jotka toteuttavat alussa määritellyt toiminnot.

Opinnäytetyön alkaessa ei ollut tekijällä kokemusta Visual Basic-ohjelmointikielestä, eikä Office-lisäosien tuottamisesta. Lisäosien ominaisuuksien tekemisen aloittaminen oli helppoa valitun kielen takia. Visual Basicin oppimisen nopeus ja helppous nopeutti ominaisuuksien valmistumista, kun merkittävää osaa kehitysajasta ei tarvitse käyttää työkalujen opiskeluun. Ominaisuuksien luominen sujui ketterästi, sillä ne voitiin testata heti, jolloin ne voitiin hylätä tai muuttaa, jos ominaisuuden toiminta ei vastannut haluttua. Projektin suurimmat ongelmat liittyivät lisäosien jakeluun. Jakelussa käytetyn Install Shield-ohjelmistopaketoijan konfigurointi osoittautui haasteelliseksi. Asennus ei tahtonut toimia yritysympäristöissä ja lisäosien päivitys kangerteli. Ongelma ratkaistiin testaamalla ja päivittämällä asennuspakettia eri ympäristöjä vasten, kunnes asennuspaketti asentui ja päivittyi ongelmitta.

Opinnäytetyön aikana suoritettu projekti tuotti halutun tuloksen, josta on mahdollista lähteä kehittämään eteenpäin projektin aikana tuotettuja lisäosia. Lisäosien kehitys jatkuu asiakasasennusten muodossa. Asiakkailta saatava palaute ohjaa kehitystä uusien ominaisuuksien muodossa, sekä toteutettujen ominaisuuksien jatkokehittämisessä.

LÄHTEET

Bellis, M 2015. The History of the BASIC Programming Language. Luettu 16.8.2016 http://inventors.about.com/od/famousinventions/fl/The-History-of-the-BASIC-Programming-Language.htm

Mack, G. 2002. History of Visual Basic. Luettu 16.8.2016 http://www.ojodepez-fanzine.net/network/qbdl/history_of_visual_basic.html

Microsoft. 2016, Architecture of VSTO Add-ins. Luettu 16.9.2016 https://msdn.microsoft.com/en-us/library/bb386298.aspx

Microsoft. 2017, Custom document properties. Luettu 5.1.2017 https://msdn.microsoft.com/en-us/library/office/microsoft.office.interop.word._document.customdocumentproperties(v=office.14).aspx

Microsoft. 2017, Custom XML. Luettu 5.1.2017 https://msdn.microsoft.com/fi-fi/library/bb608612.aspx

Microsoft, 2016. Object-Oriented Programming (Visual Basic). Luettu 16.8.2016 https://msdn.microsoft.com/en-us/library/mt656687.aspx

Microsoft, 2017. Programming VSTO Add-Ins .Luettu 20.3.2017 https://msdn.microsoft.com/en-us/library/bb157876.aspx

Microsoft, 2017. Video and audio file formats supported in PowerPoint. Luettu 4.1.2017

https://support.office.com/en-us/article/Video-and-audio-file-formats-supported-in-PowerPoint-d8b12450-26db-4c7b-a5c1-593d3418fb59

Microsoft, 2016. Visual Basic language reference. Luettu 18.82016 https://msdn.microsoft.com/en-us/library/sh9ywfdk.aspx

Microsoft, 2017. WebClient class. Luettu 3.1.2016 https://msdn.microsoft.com/en-us/library/system.net.webclient(v=vs.110).aspx

Scala, 2017. Case classes. Luettu 4.1.2016 http://docs.scala-lang.org/tutorials/tour/case-classes.html

TechnologyUK, 2016. Event-driven programming http://www.technologyuk.net/computing/software-development/event-driven-programming.shtml